Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of the claims in the application:

Listing of Claims:

1. (Currently Amended) A method to maintain a hierarchy of application objects in a computer system, the method including:

automatically detecting an exit of a child application object launched by a parent application object, wherein the computer system which maintains the hierarchy of application objects, the parent application object, and a grandchild application object, launched by the child application object, remain active; ~~and a parent application object, which launched the child application object, remain extant;~~

automatically terminating the grandchild application object after the exit of the child application object;

automatically determining whether the exit of the child application object was expected;

automatically attempting restart of the child application object if the exit of the child application object was unexpected; and

automatically signaling an outcome of the attempted restart of the child application object to the parent application object that launched the child application object prior to the exit of the child application object.

2. (Previously Presented) The method of claim 1 including maintaining an application hierarchy by recording a hierarchical relationship between the parent, child and grandchild application objects.

3. (Original) The method of claim 2 wherein the maintaining of the application hierarchy includes launching the parent application object, launching the child application object responsive to a request from the parent application object, and launching the grandchild application object responsive to a request from the child application object.

4. (Original) The method of claim 1 wherein, if the restart of the child application object is successful, then communicating a restart message to the parent application object to inform the parent application object of the successful restart.

5. (Original) The method of claim 1 wherein, if the restart of the child application object is not successful, then communicating a failure message to the parent application object to inform the parent application object of the failed restart.

6. (Original) The method of claim 1 wherein the automatic detecting of the exit of the child application object includes detecting hanging of a process of the child application object.

7. (Cancelled)

8. (Original) The method of claim 1 wherein, if the restart of the child application object is successful, then creating a new process identifier for the process of the child application object.

9. (Currently Amended) A system to maintain a hierarchy of application objects, the system including:

a watchdog automatically to detect an exit of a child application object that was launched by a parent application object, wherein the system that launched the parent application object and the child application object, remains active; and

an executor to automatically [[to]] terminate a grandchild application object launched by the child application object after the exit of the child application object, to automatically ~~determining~~ determine whether the exit of the child application object was expected, to automatically attempt restart of the child application object if the exit of the child application object was unexpected, and to automatically signal an outcome of the attempted restart of the child application to [[a]] the parent application object ~~that launched the child application object~~.

10. (Original) The system of claim 9 wherein the executor is to maintain an application hierarchy recording a hierarchical relationship between the parent, child and grandchild application objects.

11. (Original) The system of claim 10 wherein the executor is to launch the parent application object, to launch the child application object responsive to a request from the parent application object, and to launch the grandchild application object responsive to a request from the child application object.

12. (Original) The system of claim 9 wherein, if the restart of the child application object is successful, the executor is to communicate a restart message to the parent application object.

13. (Original) The system of claim 9 wherein, if the restart of the child application is not successful, the executor is to communicate a failure message to the parent application object.

14. (Original) The system of claim 9 wherein the watchdog is to automatically detect the exit of the child application object by detecting a hang state for a process of the child application object.

15. (Cancelled)

16. (Original) The system of claim 9 wherein the executor, if the restart of the child application is successful, is to create a new process identifier for a process of the child application object.

17. (Currently Amended) A machine-readable medium storing a sequence of instructions that, when executed by a machine, cause the machine to:

automatically detect an exit of a child application object launched by a parent application object, wherein the parent application and a grandchild application object launched by the child application object, remain active; ~~and a parent application object that launched the child application object remain extant;~~

automatically terminate the grandchild application object after the exit of the child application object;

automatically determining whether the exit of the child application object was expected;

automatically attempting restart of the child application object if the exit of the child application object was unexpected; and

automatically signal an outcome of the attempted restart to [[a]] the parent application object that launched the child application object prior to the exit of the child application object.

18. (Currently Amended) A system to maintain a hierarchy of application objects, the system including:

first means for automatically detecting an exit of a child application object, wherein the parent application and a grandchild application object launched by the child application object, remain active; and ~~and a parent application object that launched the child application object remain extant and;~~

second means for automatically terminating the grandchild application object after the exit of the child application object, for automatically determining whether the exit of the child application object was expected, for automatically attempting restart of the child application object if the exit of the child application object was unexpected, and for automatically signaling an outcome of the attempted restart ~~of~~ to the parent application object that launched the child application object prior to the exit of the child application object.